Supporting Group Interaction among Humans and Autonomous Agents

D. Schreckenghost, C. Martin, P. Bonasso, D. Kortenkamp, T. Milam, and C. Thronesbery

NASA Johnson Space Center
TRACLabs
1012 Hercules
Houston, TX 77058
ghost@ieee.org

Abstract

An important aspect of interaction among groups of humans and software agents is supporting collaboration among these heterogeneous agents while they operate remotely and communicate asynchronously. We are developing an architecture that supports multiple humans interacting with multiple automated control agents in such a manner. We are evaluating this architecture with a group consisting of the crew of a space-based vehicle and the automated software agents controlling the vehicle systems. Such agent interaction is modeled as a loosely coordinated group because this model minimizes agent commitment to group goals and constraints while addressing a significant portion of crew and control agent group behaviors. In this paper we give background on human interaction with space-based automation. We identify related research in multi-agent autonomous architectures and single agent human-computer interaction systems. We describe our architecture design for human-software agent groups. And we identify research issues in loosely coordinated human-software groups.

Background: Human Interaction with Space-based Automation

Future manned space operations will include greater use of automation than is seen today. Automated software agents will perform difficult tasks like system control while operating mostly autonomously. Such sophisticated software agents have been referred to as immobile robots or *immobots* (Williams and Nayak, 1996) due to their computational similarity to physical robots. As automated agents like robots or immobots become more prevalent, human interaction with them becomes more frequent and routine. One particular type of group interaction that has yet to be explored greatly in the research is the interaction among multiple humans and multiple autonomous agents.

We are investigating the interaction among distributed human and control software agents when organized in loosely coordinated groups. By loose coordination we mean that (1) domain responsibilities are allocated to group members based on related but non-overlapping roles, (2) activities of group members are coordinated by means of a pre-built, high-level, centralized plan that manages shared resources, and (3) unplanned actions conducted by group members are prevented from interfering with the ongoing actions of other group members. We have chosen to investigate loosely coordinated groups because they represent an approach that minimizes agent commitment to group goals and constraints while addressing a significant portion of crew and control agent group behaviors. The simplifying assumptions of loose coordination (i.e., centralized planning, no dynamic reconfiguration of teams) make multi-human/multi-agent interaction realistic but tractable in a complex domain like space. We have observed that the need for human-agent teams that can be dynamically reconfigured is not common in space-based operations because both human and control agents have highly specialized skills that prevent flexible reconfiguration of human-software teams. We do expect, however, to extend our approach in later years to support more tightly coordinated human-software teams formed for the purpose of cooperative, traded control activities that include joint task execution.

An important aspect of interaction among loosely coordinated groups of humans and software agents is supporting collaboration among these heterogeneous agents while they operate remotely and communicate asynchronously. For example, in the space domain, the crew should be able to monitor and control autonomous operations from any location at the site and with only occasional intervention. This requires the crew be able to quickly form an integrated view of distributed control without continuously monitoring control data. It also requires the crew be able to command control systems from anywhere inside or nearby outside the space site. And, at the crew's discretion, they must be able to override autonomous control in response to system anomalies and mission opportunities.

The crew should be supported in interleaving group activities like monitoring and control operations with non-group manual activities like performing science tasks. Since nominal operations for control agents will be mostly autonomous, the crew typically will spend their time on non-group tasks. Occasionally, however, the crew must respond to unusual situations requiring more active intervention. In effect, the crew is “on call” to handle these situations. This requires assistance in handling interruptions and managing increased workload when “called in”.

Crew located throughout the site should be able to collaborate with other members of the group (other crew as well as automated control agents) distributed throughout site. This requires adapting the standard interaction protocols used in manned space operations today to address what information to communicate, and how and when to notify remote crew of important events and system status. These protocols also must accommodate asynchronous communication. Crew participating in the collaboration must be able to make control decisions jointly. When more than one crew member is commanding, it is necessary to prevent conflicting control commands and to assist in reconfiguring automation for manual intervention, if needed.

Related Work

Very little previous research has focused explicitly on interaction among multiple humans and multiple semi-autonomous software agents (e.g. control systems). However, to achieve these goals we can leverage existing research that focuses on coordination and distributed collaboration among multiple software agents as well as existing research that focuses on interaction between individual humans and software agents. This existing research can be applied to support multi-human/multi-agent collaboration. We examined a number of implemented systems that helped inform our initial architecture design.

Previous research has explored several interaction models, algorithms, and system characteristics that support different types of collaboration capabilities among multiple entities or between individual humans and software. Much previous work has focused on multi-agent interaction, including coordination and collaboration in distributed multi-agent systems (Jennings, 1996; Jennings et al., 1998; Lesser, 1998; So and Durfee, 1998). Previous research has also addressed individual human/agent collaboration needs including the development of “advisable” agents that incorporate a user’s preferences for when to ask for permission or consultation for given behaviors (Myers and Morley, 2001) as well as the development of reminding systems that consider whether or not to issue a reminder based on the importance of a task and the likelihood that it will be forgotten (McCarthy and Pollack, 2001).

In addition, several previously developed systems have implemented the capability to adjust the level of human intervention in the actions of autonomous systems (i.e., adjustable autonomy) (Dorais et al., 1998; Kortenkamp et al., 2000; Scerri et al., 2001). The dimensions of agent autonomy and adjustable autonomy continue to be explored (Barber et al., 2001; Castelfranchi, 1995; Hexmoor, 2001; Luck and d'Inverno, 2001). Although no common view of agent autonomy has been reached in the research community, the exploration of these concepts has helped us better understand the complex relationship between humans and autonomous agents. Further examining these relationships, discourse models supporting a shared context for human-agent interaction and mixed-initiative planning have been developed by COLLAGEN (Rich and Sidner, 1998) and TRIPS (Ferguson and Allen, 1998). In addition many researchers have focused on providing specifications for agent-to-agent discourse in multi-agent systems (Bradshaw et al., 1997; Labrou et al., 1999). These specifications, ranging from message syntax to conversation policies, provide a foundation that supports interaction among groups of agents.

In particular, one very successful implementation of interaction between humans and software agents has been demonstrated in the Electric Elves system (Chalupsky et al., 2001; Pynadath et al., 2000). In this system, proxy agents for each person in an organization perform organizational tasks for their users including (among other things) monitoring the location of each user, keeping other users in the organization informed, and rescheduling meetings if one or more users is absent or unable to arrive on time. The Electric Elves system does incorporate multiple humans and multiple software agents; however, each human interacts primarily with the capabilities of his or her own proxy (or with non-autonomous software accessed through the proxy). The Electric Elves architecture is relevant and useful for informing the design of our software architecture; however, it does not fully address our requirements for support agents (proxies) to act as mediators and/or enablers for humans to interact with yet a third class of agents: autonomous control systems.

Approach: Distributed Crew Interaction Architecture

We are developing a distributed crew interaction (DCI) architecture (see the diagram below) to assist the crew in remotely interacting with automated control agents for crew life support. Thus, this architecture must support interaction among groups of human and software agents. A key element of this design are collaborative agents that assist humans and automated control agents in working together. These Collaborative Agents can fulfill a variety of roles to aid collaboration, including (1) an aide or stand-in for the crew, (2) an augmentation of crew capabilities, or (3) a regulator or critic of crew actions.

The *Crew Proxy* is a collaborative agent central to the DCI architecture. Each crew member has a Proxy to represent his interests and concerns. The Crew Proxy is a coordinator of services that can be customized for individual crew members. Services available to the Crew Proxy include the following:

- *Notification Service*: uses the information about crew state, roles, and preferences to determine if an operational event is of interest to a crew member and, if so, how to inform the crew member.
- *Task Status Service*: provides activity tracking and plan management capabilities for use by both the crew and the autonomous control agents affected by crew activities.
- *Crew Location Service*: provides crew location information for use (1) by the Task Status Service in tracking the completion status of crew activities, (2) by the Notification Service in determining how to notify the crew of events, and (3) by the User Interface in customizing the presentation of information.
- *Command and Authorization Service*: supports the crew in remotely interacting with and controlling the life support systems by (1) determining if the crew is authorized to command (i.e., access control), (2) resolving authorization conflicts when more than one crew member is interacting with the life support systems, and (3) reconfiguring both the automation and user interface in preparation for commanding.
- *Interactive Event Service*: assists the crew in interactively defining temporary, new operational events and controlling automated monitoring for these events.
- *Interactive Procedure Service*: assists the crew in temporarily modifying standard operating procedures executed by the automated control software.
- *Interruption Handling*: consists of extensions to the other services as well as a new Interruption Handling Service that manages concurrent crew activities.

The DCI architecture also includes *Control Assistants* that aid the crew in interacting with automated control agents. These Control Assistants inform all Crew Proxies about control events affecting the life support systems, including events resulting from both automated control actions and crew control actions. We have defined two types of Control Assistants needed for crew interaction. The Event Detection Assistant detects and broadcasts significant operational events including anomalies that occur in the life support systems. The Crew Error Detection Assistant detects conditions indicating the crew has taken an action with potentially adverse effects on the environment.

The DCI architecture will be evaluated by integrating it with Autonomous Control Agents developed for a space applicatoin. The types of Autonomous Agents we will use include the following:

- *Life support control agents* based on the 3T control architecture (Bonasso et al., 1997a) include crew air revitalization (Schreckenghost et al., 2001) and water recovery (Bonasso, 2001).
- *Crew activity planning agents* (Schreckenghost and Hudson, 2001) that assists the crew in building and monitoring crew activity plans that interact with autonomous control.
- *Robotic mobile monitors* (Kortenkamp et al., 2002) that perform inspection, monitoring and sensing tasks.

The initial prototype under development will be integrated with the water recovery control agent and tested by control engineers supervising the Advanced Water Lab at JSC.

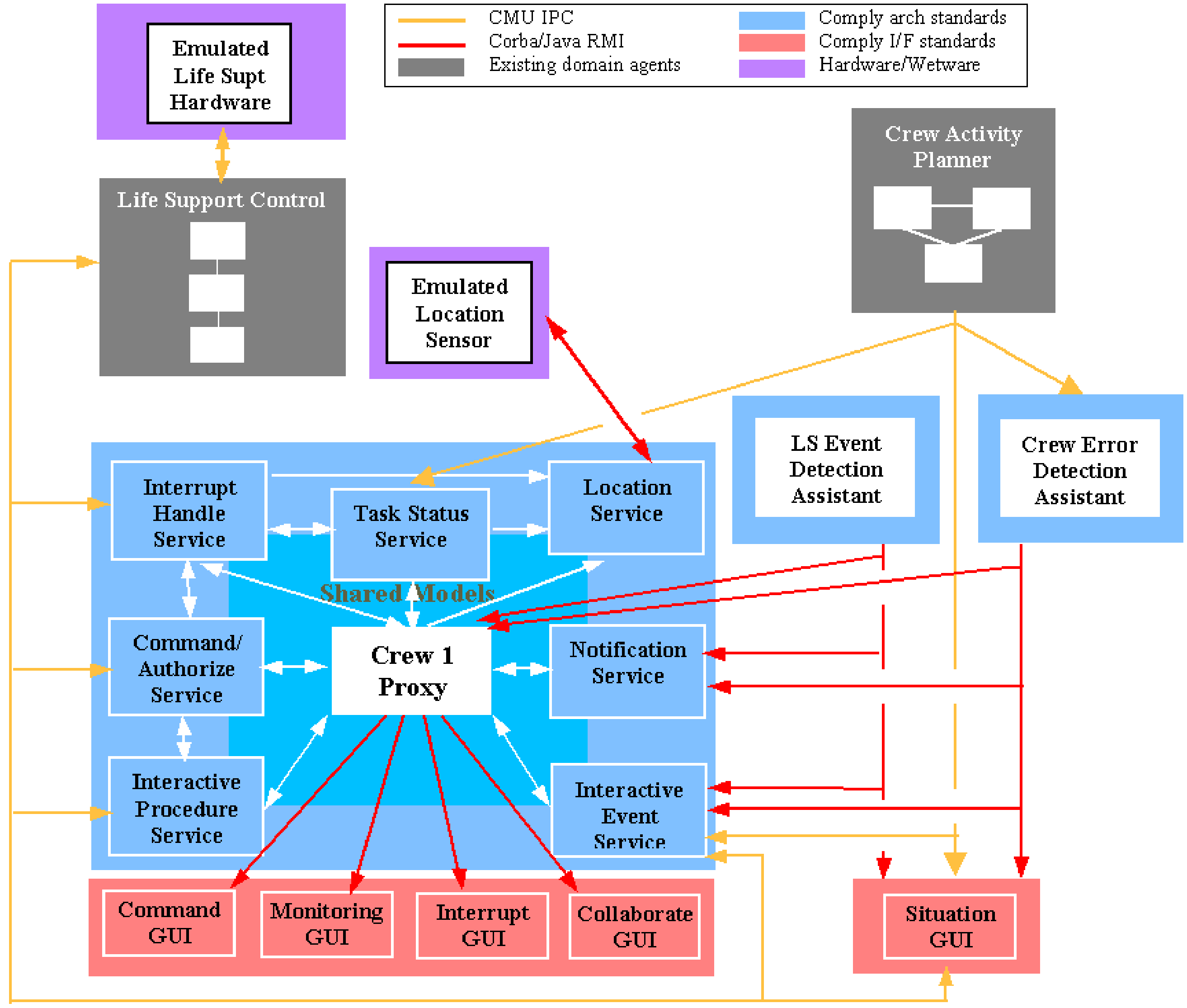

Diagram: Distributed Crew Interaction Architecture

Research Issues: Loosely Coordinated Groups

We are currently investigating the following issues for loosely coordinated groups:

Crew Intervention and Commanding: When situations arise that fully autonomous operations cannot address, it is necessary to support some level of crew interaction and intervention with the autonomous control software. The autonomous control agents developed for life support are capable of *adjustable levels of autonomy* (Bonasso et al., 1997b; Kortenkamp et al., 2000). The Crew Proxies will assist the crew in determining how to adjust the level of autonomy to support standard manual procedures (e.g., maintenance activities such as filter replacement or sensor calibration). We are investigating a control ontology that describes the degree to which autonomous response must be suspended to permit different types of manual intervention. We will use this ontology to determine how to automatically reconfigure the automated control agents when the crew performs standard manual procedures.

Applying this technique for adjusting the level of control autonomy when the crew are distributed throughout a facility requires the ability to *remotely command* life support systems. The Crew Proxies will assist the crew in remotely commanding by reconfiguring automation for manual actions, and by providing computer-based interfaces for manual execution of control procedures. The Proxies also will authenticate remote users and resolve conflicting commands from crew members at different locations.

Dynamic Event Notification: Groups for system control are formed by assigning related operational roles to both the crew and the autonomous control agents. Since role assignments are specific to a single agent, crew-specific *event notification* is needed to inform the crew of operational events and control actions taken by other members of the group (human and automated agent) that are important to their assigned roles. While it may be possible for both the crew and the autonomous control agents to fulfill the same role, only one agent is assigned to a role at a time. These roles can be dynamically re-assigned by the autonomous planner or by the crew. Changes in crew role alter the information and commanding requirements of the crew. The Crew Proxy for each crew member knows *what events* its user should see and *how to inform* its user of these events. Whether its user is interested in an event is determined based on the roles its user currently fulfills. For distributed, remote operations, knowledge of how to inform its user will depend upon the user's current state (i.e., whether a crew member is online and what type of computing platform the crew member is using) and the user's preferred means of being informed (e.g., audio, graphical, email, etc.).

The notion of pre-defined event detection and notification can be extended to include *interactive event monitors* that assist the crew in dynamically defining temporary monitors for operational changes in response to unusual situations or operations. These temporary monitors address the information needs of a subset of the group responding to the unusual situation. When a temporary event is detected, only those agents identified as interested agents are notified.

Group Plans for Human and Software Agents: The coordination of crew and autonomous control agent activities will be based on a centralized high-level group activity plan to prevent conflicting commands, to avoid over-subscribed agents, and to assist handover between manual and automated tasking. This centralized planning capability must be able to react to contingencies in the control situation by automatically replanning. Such dynamic replanning requires the ability to detect when tasks in the plan are completed successfully or when they fail to complete, as well as the ability to adjust crew and software agent roles in response to contingencies. To assist in such closed-loop planning, the Crew Proxy will automatically track the completion of the manual activities its user performs. It will do this using both direct evidence obtained through computer-mediated manual tasks, and indirect evidence based on crew location and planned tasks. The Proxies also will assist in coordinating distributed human and software agents by reminding their users of pending tasks and task deadlines, and notifying them when their tasks change due to an automatic replan. As our mechanisms for supporting this type of group coordination mature, future extensions to this work can investigate the impact of increasingly distributed or locally reactive planning at increasingly higher levels of abstraction.

Conclusions and Future Work

We have described our approach for supporting loosely coordinated groups of human and autonomous software agents. We have identified the following research issues associated with such group models:

- Assisting humans in adjusting the level of autonomy in the automated software agent,
- Managing conflicting or interfering commands when multiple, distributed agents are authorized to command remotely ,
- Dynamically notifying an agent of control events and the actions of other agents based on the agent's role, state (such as location), and preferences,
- Defining new events and associated event monitors on-the-fly in response to novel situations, and

- Tracking the completion of manual activities for the purposes of closed-loop planning.

We have illustrated how this approach can be applied to improve manned space operations. We believe, however, that this work is relevant to other domains where humans must work together with complex autonomous agents. These domains include autonomous robotics, automation for the care of the elderly, and automated process control (including nuclear power).

We are currently implementing the Crew Proxy and its core services. We will evaluate the proxy software by using it to assist control engineers managing the autonomous Water Recovery System (WRS) at Johnson Space Center (JSC). The WRS automated control system has operated 24/7 for over a year. Control engineers who also have additional full-time job responsibilities check in on this system a few times a day to determine if a problem requiring human intervention has occurred. Three engineers are on-call. Each week a different one of these three has primary responsibility to supervise the automation. When a problem occurs, the primary engineer attempts to fix the problem if he is available. If he cannot fix the problem or is unavailable, the backup engineers take responsibility. The proxies for these control engineers will notify engineers of problems based on currently assigned role, will manage new human tasks resulting from these problems, and will use engineer availability and location to help re-allocate tasks to backup engineers.

Future work on this project will focus on extending our multi-human/multi-agent architecture to support complex types of interaction. Support for complex interaction includes such topics as:

- *Interruption Handling*: support for (1) determining if an agent should be interrupted, and how intrusive interruption should be, (2) marking completion status of interrupted activities, and (3) assisting an agent in managing multiple contexts and concurrent threads of activity.
- *Joint Task Execution*: support when the human and automation work together to accomplish a shared goal (e.g., joint human-robot tasks), such as (1) interactively adjusting autonomy, and (2) providing team models for coordinating joint tasks.

References

Barber, K. S., Martin, C. E., Reed, N. E., and Kortenkamp, D. 2001. Dimensions of Adjustable Autonomy. In *Advances in Artificial Intelligence: PRICAI 2000 Workshop Reader. Four Workshops held at PRICAI 2000, Melbourne, Australia, August/September 2000. Revised Papers*, vol. LNAI 2112, Kowalczyk, R., Loke, S. W., Reed, N. E., and Williams, G., Eds. Berlin: Springer, 353-361.

Bonasso, R. P. 2001. Intelligent Control of a NASA Advanced Water Recovery System. In *Proceedings of the ISAIRAS*.

Bonasso, R. P., Firby, J. R., Gat, E., Kortenkamp, D., Miller, D. P., and Slack, M. G. 1997a. Experiences with an Architecture for Intelligent, Reactive Agents. *Journal of Experimental and Theoretical Artificial Intelligence* 9: 237-256.

Bonasso, R. P., Kortenkamp, D., and Whitney, T. 1997b. Using a Robot Control Architecture to Automate Shuttle Procedures. In *Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence (IAAI)*, 949-956. Providence, RI.

Bradshaw, J. M., Dutfield, S., Benoit, P., and Woolley, J. D. 1997. KAoS: Toward an Industrial-strength Generic Agent Architecture. In *Software Agents*, Bradshaw, J. M., Ed. Cambridge, MA: AAAI Press/ The MIT Press, 375-418.

Castelfranchi, C. 1995. Guarantees for Autonomy in Cognitive Agent Architecture. In *Intelligent Agents: ECAI-94 Workshop on Agents Theories, Architectures, and Languages*, Wooldridge, M. J. and Jennings, N. R., Eds. Berlin: Springer-Verlag, 56-70.

Chalupsky, H., Gil, Y., Knoblock, C. A., Lerman, K., Oh, J., Pynadath, D. V., Russ, T. A., and Tambe, M. 2001. Electric Elves: Applying Agent Technology to Support Human Organizations. In *Proceedings of the Innovative Applications of Artificial Intelligence*. Seattle, WA.

Dorais, G. A., Bonasso, R. P., Kortenkamp, D., Pell, B., and Schreckenghost, D. 1998. Adjustable Autonomy for Human-Centered Autonomous Systems on Mars. In *Proceedings of the Mars Society Conference*.

Ferguson, G. and Allen, J. F. 1998. TRIPS: An Integrated Intelligent Problem-Solving Assistant. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*. Madison, WI.

Hexmoor, H. 2001. Situated Autonomy. In *Intelligent Agents VII: Agent Theories Architectures and Languages*, Castelfranchi, C. and Lesperance, Y., Eds. Berlin: Springer, 349-350.

Jennings, N. R. 1996. Coordination Techniques for Distributed Artificial Intelligence. In *Foundations of Distributed Artificial Intelligence*, *Sixth-Generation Computer Technology Series*, O'Hare, G. M. P. and Jennings, N. R., Eds. New York: John Wiley & Sons, Inc., 187-210.

Jennings, N. R., Sycara, K., and Wooldridge, M. 1998. A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems (AAMAS-98)* 1(1): 7-38.

Kortenkamp, D., Keirn-Schreckenghost, D., and Bonasso, R. P. 2000. Adjustable Control Autonomy for Manned Space Flight. In *Proceedings of the IEEE Aerospace Conference*. Big Sky, MT.

Kortenkamp, D., Schreckenghost, D., and Martin, C. E. 2002. User Interaction with Multi-Robot Systems. In *Proceedings of the NATO Multi-Robot Systems Workshop*. Naval Research Laboratory, Washington, D.C.

Labrou, Y., Finin, T., and Peng, Y. 1999. Agent Communication Languages: The Current Landscape. *IEEE Intelligent Systems* 14(2): 45-52.

Lesser, V. R. 1998. Reflections on the Nature of Multi-Agent Coordination and Its Implications for an Agent Architecture. *Autonomous Agents and Multi-Agent Systems (AAMAS-98)* 1(1): 89-111.

Luck, M. and d'Inverno, M. 2001. Autonomy: A Nice Idea in Theory. In *Intelligent Agents VII: Agent Theories Architectures and Languages*, Castelfranchi, C. and Lesperance, Y., Eds. Berlin: Springer, 351-353.

McCarthy, C. E. and Pollack, M. E. 2001. *A Plan-Based Personal Cognitive Orthotic for the Elderly*, In Progress, University of Pittsburg/University of Michigan.

Myers, K. L. and Morley, D. N. 2001. Directing Agent Communities: An Initial Framework. In *Proceedings of the IJCAI-2001 Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents*, 81-88. Seattle, WA.

Pynadath, D. V., Tambe, M., Arens, Y., Chalupsky, H., Gil, Y., Knoblock, C. A., Lee, H., Lerman, K., Oh, J., Ramachandran, S., Rosenbloom, P. S., and Russ, T. A. 2000. Electric Elves: Immersing an Agent Organization in a Human Organization. In *Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents: Human in the Loop*, 150-154.

Rich, C. and Sidner, C. L. 1998. COLLAGEN: A Collaboration Manager for Software Interface Agents. *User Modeling and User-Adapted Interaction* 8(3-4): 315-350.

Scerri, P., Pynadath, D. V., and Tambe, M. 2001. Adjustable Autonomy in Real-World Multi-Agent Environments. In *Proceedings of the Autonomous Agents*, 300-307. Montreal, Canada.

Schreckenghost, D. and Hudson, M. B. 2001. Automation in Context: Planning Manned Space Exploration Activities. In *Proceedings of the ISAIRAS*.

Schreckenghost, D., Malin, J., Thronesbery, C., Watts, G., and Fleming, L. 2001. Adjustable Control Autonomy for Anomaly Response in Space-based Life Support Systems. In *Proceedings of the IJCAI-2001 Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents*. Seattle, WA.

So, Y.-p. and Durfee, E. H. 1998. Designing Organizations for Computational Agents. In *Simulating Organizations*, Prietula, M. J., Carley, K. M., and Gasser, L., Eds. Menlo Park, CA: AAAI Press / The MIT Press, 47-66.

Williams, B. C. and Nayak, P. P. 1996. Immobile Robots - AI in the New Millennium. *Artificial Intelligence* 17 (3).